

TICOM II: The Internal Control Language

Andrew D. Bailey, Jr.  
University of Minnesota

James Gerlach  
Purdue University

R. Preston McAfee  
Purdue University

Andrew B. Whinston  
Purdue University

## Abstract

The importance of the review, evaluation, and testing of accounting internal control systems in an audit environment is well established. The enactment of the Foreign Corrupt Practices Act of 1977 further emphasized the importance of accounting internal control systems by making it illegal not to maintain adequate systems of this type. These facts, coupled with interest in the possibility of separate audit opinions addressing the adequacy of a management's system of accounting internal controls, should be sufficient to establish the importance of developing more effective means of documenting, studying, and evaluating such systems.

This paper presents an internal control language, TICOM II, which is based in mathematical logic. Manual and/or computerized office information systems can be described in the TICOM II language. The resulting description is maintained on a computer. Analysis of the description is then based upon the interaction of the auditor, who poses queries as to the system characteristics, and computer software analysis of those queries. The benefits of this approach are derived from the theoretical properties of its mathematical logic basis and from the use of the computer's computational power.

## Introduction

The focus of this paper is on the study and evaluation of internal accounting controls in large organizations. Control within an operation and the study of the "coordinate methods and procedures" necessary to attain it date back many thousands of years (Griffith, 1963). Historians can undoubtedly weave a most interesting thread through time and events that would link these early considerations of control to today's business or governmental organization. Recent events have considerably increased the potential for conflict on matters of internal control, thus the greater interest in the study of such matters.

If a single event can ever be considered crucial in matters as complex as those considered here, it is the discovery or rediscovery of widespread illegal and illicit payments by U.S. business firms doing business in foreign countries. Revelations of this nature are not now new, nor have they been in the past, nor is it a wholly new phenomenon that accountants have been identified with the issues. What is perhaps new is the level of governmental response and the resulting legislation in the form of the Foreign Corrupt Practices Act of 1977 (FCPA'77). As the FCPA'77 bears on the accounting profession, it is at first glance an apparent vindication of the profession. It requires public companies to "make and keep books, records and accounts, which in reasonable detail, accurately and fairly reflect the transactions of the assets of the issuer," and further to

devise and maintain a system of internal control sufficient to provide reasonable assurance that

- (i) transactions are executed in accordance with management's general and specific authorizations;
- (ii) transactions are recorded as necessary:
  - (I) to permit preparation of financial statements in conformity with generally accepted accounting principles or other criteria applicable to such statements, and
  - (II) to maintain accountability for assets;
- (iii) access to assets is permitted only in accordance with management's general or specific authorization; and

- (iv) the recorded accountability for assets is compared with existing assets at reasonable intervals and appropriate action is taken with respect to any differences. (DH&S, 1979)

These words were adapted from Statement on Auditing Standards (SAS) Section 320 of the American Institute of Certified Public Accountants (AICPA), but they now have the force of law with corresponding criminal penalties including imprisonment for those responsible for a failure to adhere to the law (AICPA, 1972, Section 320).

If we ignore certain nontrivial problems with the legislation as written, such as its lack of a specified materiality criterion, the FCPA'77 would vindicate a long-standing professional position with respect to the importance and objectives of internal controls. Unfortunately the accounting profession identified the issue but not the means of accomplishment. This has undoubtedly come as a surprise to many of the uninitiated who assumed that these matters were well in hand and that failures of internal control somehow demonstrated the incompetence, or the tacit knowledge and approval of, the accountant and/or auditor. In fact the failure of such systems may be due to the complexity of large organizations and fundamental limitations in systems development and auditing concepts and techniques. The state of the art was and is simply not well enough advanced to assure conformity with legislation such as the FCPA'77. We believe that complexity of controls in today's large organizations preclude adequate systems description and analysis without the aid of computers.

The profession, through its members, quite properly points out that while the words of the FCPA'77 and the Statement on Auditing Standards may be the same, the objectives differ substantially. The AICPA standards and related publications were primarily intended to support an audit function within which internal controls were simply one aspect bearing on the potential for material error in financial statements. The auditor's evaluation of internal controls was directed at assessing the potential for reliance on them in order to devise the tests necessary to substantiate that no material error existed in reported financial statements. It was never the intent of the auditor to render an opinion on the "overall quality" of internal controls themselves. This point is made eminently clear throughout the statements on auditing

standards. What perhaps was not so clear was that the auditor's current technology would not support such an opinion should the objective of the audit include a separate opinion on internal control.

Whatever the real or perceived limitations faced by accountants and auditors, we believe that the FCPA'77 has laid the foundation on which we can expect the SEC to place a structure which will include statements by management concerning internal control and some form of audit opinion as to the adequacy of internal controls. Recent actions by the SEC clearly demonstrate that this is their intent. That the auditing profession recognized this intent is evident by Reporting on Internal Control, which not only proposes a new opinion on internal control without restrictions as to distribution but explicitly addresses the standard's relationship to the FCPA'77 (AICPA, 1972, SAS No. 30). The relationship is tenuous in that the new standard clearly recognizes and incorporates the limitations of current audit technology in casting the opinion statement on internal control. Whether this proposal will satisfy the SEC and other responsible agencies remains to be seen.

The accounting profession has available only two avenues along which to attack the problems brought to a head by the enactment of the FCPA'77. First, it can attempt to educate those who wish to impose this new responsibility on the profession about the limitation of the current state of the art of systems design and audit technology. The current withdrawal of the suggested reporting and auditing statements with respect to internal control by the SEC may indicate some success with this tactic. In our opinion this will be only a temporary and/or temporizing success. The second approach is to expand the profession's understanding of the development and analysis of internal control systems.

This paper possesses elements of both lines of argument. While the primary thrust of the work relates to improving design and analysis methodologies, we also believe that unrestricted resolution of the control problem is not conceptually viable. This point may be self-evident; however, we have provided a theoretically sound statement as to this fact (Bailey et al., 1980a). The major effort of this paper involves an attempt to make better use of the computer in the study of internal control systems. While our orientation is in the first

instance based on audit issues, it bears on design considerations as well. In fact, in the broader view of internal control matters, the benefits of TICOM II may be more apparent in its potential as a design and control tool.

#### Two "Views" of Internal Control

After reviewing the literature concerning the audit techniques associated with internal control, we conclude that variants of two approaches are currently being implemented: a cycles approach and a transactions and account classification approach. The cycles approach is clearly stated in the AICPA Report of the Special Advisory Committee on Internal Control (1980) and by Arthur Andersen & Co. in their publication A Guide for Studying and Evaluating Internal Accounting Controls (1978). The transactions and account classification approach is well illustrated by the Deloitte Haskins & Sells publication Internal Accounting Control, An Overview of the DH&S Study and Evaluation Techniques (1979).

Briefly, "a cycle consists of transactions reflecting related activities that can be conveniently grouped together" (AA & Co., 1978). The following cycles have been proposed:

<u>Cycles</u>	<u>Examples of Areas Involved</u>
Revenue	Order entry, credit, sales, shipping, accounts receivable, cash receipts
Expenditures	Purchasing, receiving, accounts payable, payroll, cash disbursements
Production or conversion	Inventory, property accounts, labor cost of sales
Financing	Capital stock, debt, dividends, interest
External financial reporting	Preparation of financial statements, adjustments, disclosures

Except for the last cycle, each clearly highlights the flow of resources through the business. The last cycle is necessary to allow for the end-of-period preparation of financial reports based on the other four cycle activities.

The transaction and account classifications approach attempts to follow lines of organizational responsibility rather than cutting across these lines as is purposely done in the cycles approach. Example categories of transactions and accounts are Cash receipts, Cash disbursements, Cash balances, Sales and trade receivables, Purchase and trade payables, Payroll, Securities, Inventories, Property, Other assets and liabilities, Journal entries and general ledger, Other control considerations, and External financial reporting (DH&S, 1979).

Proponents of these "competing" approaches each propose strong arguments as to why their approach is superior to the other. It is not our purpose to debate these issues. In fact, the more we consider the issue, the less relevant the choice becomes within the context of our research. Suffice it to point out the obvious: each approach utilizes a different view of the same business entity. This point is most obvious in that within each approach are the components of the other; for example, from the transactions and account approach we may knit together the revenue cycle--sales and trade receivables, cash receipts. Proponents of both approaches also emphasize the need to establish controls along the general transactions flow of authorization, approval, execution, recording, custody of assets, and reconciliation of the recorded accountability for assets with the existing assets.

The system of description and evaluation methods proposed by this research can be employed under either of the "views" of internal control discussed. The analysis methods do not depend on a cycle viewpoint. The techniques provide different viewpoints that may be needed to support data collection. For example, a functional view may permit easy data collection within a department such as purchasing. Within purchasing an even more individualized view would permit each person freedom in describing his activities on behalf of purchasing. Note that an individual may carry out tasks that overlap organizational units. It may be useful from a purely informational point of view to organize the description so that all the tasks an individual performs are available. The ability to edit, modify, and move descriptive text is a useful feature for the auditor. At this stage of development we are relying on flowcharts for the initial description, but all of the written techniques employed contribute to refinement of the system description,

for example, internal control questionnaires and descriptors such as those employed by DH&S (1979).

#### Model System

In order to develop a demonstration case it was necessary to make a number of simplifying choices. As it was neither our intent nor our focus to develop new accounting control systems, we chose to accept an existing representation of such a system. Robertson, in his text, Auditing (1979), presents a series of accounting control systems flowcharts within which the following are included: sales processing system, accounts receivable processing system, cash receipts processing system, purchasing system, inventory system, cash disbursements system, and payroll processing system.

These flowcharts do not represent a complete accounting control system, nor are they as highly detailed a description as is potentially available within an organization; however, they are sufficient for our purposes. Note that the listing of subsystems conforms reasonably closely to the DH&S transactions and account classification "view" of the control process. Also, by combining subsystems listed by Robertson we obtain a cycles "view" of the control process similar to the American Accounting Association (AAA) and the AICPA. For example, combining the sales, accounts receivable, and cash receipts systems, we obtain a basic revenue cycle; combining purchasing and cash disbursements or payroll and cash disbursements, we obtain the basic expenditures cycle.

The review and evaluation of internal control required by GAAS Field Standard 2 could be based on the flowchart description above, internal control questionnaires, descriptive statements, decision table analysis, and other means thought appropriate in the circumstances. There would follow from this review and evaluation a proposed audit program. The current state of the art in these matters is limited in part by human analytic capabilities. This condition limits the complexity, depth, and breadth of the review and evaluation process. One means of expanding the auditors review, evaluation, and testing potential involves computerization of parts of the process. In this way



the complexity, depth, and breadth of analysis can be extended by means of machine analytic capabilities.

A computer-based description of internal control will provide several immediate benefits to the auditor with little change in the manner in which audit activity currently occurs. For example, once computerized, the description can be updated with ease, avoiding the current redrafting of paper-based descriptions. Further, the computerized analytic capabilities can be utilized to provide an analysis of internal consistency of a change or proposed change in the control system. Current internal control questionnaires and decision table analysis can be used to develop individual queries of the system description or, in many cases where a decision table analysis of potential weaknesses exists, the decision table analysis can itself be computerized as a set of automated queries on the updated internal control system description. Automation at this level will permit a greater depth and breadth of study along traditional lines. Research could conceivably achieve the application of artificial intelligence to the complete audit process, but its goals have been much more limited and, we believe, more nearly attainable. Basically we intend to develop a means of computerizing the internal control description of an organization initially described by traditional means.

Another significant application area is in systems design and the emerging field of Office Information Systems (OIS). The potential for significant productivity gains through automated OISs has stimulated considerable thought and research. The primary focus in the OIS area has been on how to design efficient and/or streamlined office systems (Bailey et al., in press). Issues of internal control have not been given serious consideration. It is our view that there is a considerable interface between the office automation field and the role and responsibility of the auditing profession. The auditor's concern for control must carry over to the automated environment. Thus the design process for the office system must involve the auditor to ensure that the requirements of internal control are preserved. The representation techniques we propose for documenting the information and decision flow within internal control systems should be valuable tools supporting both the auditor's involvement in the design process and indeed as the

beginning blueprint for the conversion to the automated environment. The large investment required by the FCPA to document the internal control system may be recouped by using this documentation as the basis for design of the automated office.

Any benefits to be derived, immediate or long-term, are dependent upon our ability to describe internal control in a language capable of computer implementation and susceptible to mathematical manipulation. A number of potential internal control languages have been considered in some depth. We believe that a logic-based language such as TICOM II provides a suitable basis for a formal language representation of internal control.

#### TICOM II:

##### Introduction

The primary purpose of this paper is to present an internal control language, designated TICOM II. The language was designed to support the description specification of accounting internal control systems. The modeling technique employed permits the description of the control structure and information needs of the organizational units within the organization. The resulting model formulation facilitates mechanical analysis of the internal control structure and information flows of the internal control system. A brief discussion on the analytic capabilities of TICOM II follows the presentation of the internal control language.

The TICOM II language is strictly specified in order to be unambiguous and to facilitate mechanical recognition and interpretation. The language constructs and terminology chosen relate closely to the fundamental concepts of internal control and systems design. It was our intent to make the language rich enough to support system descriptions that are easily readable while avoiding the recognition and interpretation difficulties inherent in natural languages.

Most important, the TICOM II language is based on a mathematical logic. The particular mathematical logic selected is sufficiently powerful to enable deductive reasoning concerning a wide range of internal control issues. It has been theoretically demonstrated that

such a deductive reasoning process is computer implementable (Bailey et al., 1980b); thus, we can rely on the memory capacity and computational speed of the computer.

The resulting language and supporting software is intended as an aid, not a substitute, for auditor judgment. The process of review and analysis will still be dependent upon the auditor for direction and resolution. In order to facilitate this interaction, a query language is proposed.

#### Research Issues

The process of deciding whether the internal control procedures described by the TICOM II language are in compliance with internal control requirements is a special reasoning process on a knowledge base that is formulated from the system description. A number of researchable issues can be identified.

First, the adequacy of the proposed language to describe an actual office environment must be established. We have adopted initially two criteria as a means of assessing proposed languages: the ability of the language to incorporate all of the essential factors described by traditional flowchart methods, and the ability to apply deductive reasoning processes to the language description appropriate to internal control analysis. While a general reasoning capability of an artificial intelligence nature would be desirable, it is our intent to consider only a limited type of deductive reasoning related to whether the modeled system can enter a given state for a particular transaction.

A second set of issues relates to the practical efficiency of such a descriptive and reasoning system. The analysis of language structures by means of computer software requires substantial computational time unless the language and queries involved are of a limited and well-specified type. In general the combining problems of language analysis are beyond the limits of existing computational capabilities, but because our concerns are in a limited area of description and analysis, we believe that relevant issues can be analyzed in a computationally economic manner.

A third matter of some concern, but beyond the scope of this paper, relates to the manner in which the auditor will interact with and be

affected by the existence of such techniques. These same issues can be addressed at a level removed from the auditor when a firm decides to adopt such methods in the design and implementation of automated office information systems.

This paper concentrates on the immediate problems of language design and system analysis based on the descriptive language.

#### Components

TICOM II is composed of four major components which facilitate the study, evaluation, and testing of internal control systems. The first component is a computer-acceptable language suitable for modeling OIS behavior. The language permits individuals or organizational units to describe independently their own activities. Activities are defined in terms of file manipulations, document manipulations, validation tests, and document transfers. In general, the significant characteristics of internal control are describable with these fundamental operations. Currently, internal control descriptions are commonly executed by flowcharting and narrative descriptions of the actions of the organizational units. This approach is exemplified by the flowchart description of a cash disbursements system in figure 1. The auditor's study of the internal control structure of the organization may be based on these charts and supporting narratives. The language we propose is based on the actual procedures and objects found in flowcharts and narrative descriptions of accounting internal control systems.

The second component of TICOM II is a synthesis system which combines the modularized procedure descriptions into a unified description of the organization. Module description may take place at the subsystem level (such as that illustrated in Figure 1), on a functional level, or even at the level of the individual operator. In this way TICOM II permits decentralization of the description process. This modeling strategy enables individuals to describe familiar duties and allow the machine to perform the tedious task of connecting the individual system components. These components are interconnected through data transfer represented by document transfers and file storage and retrieval operations. This stage of analysis will also detect inconsistencies in the individual system component descriptions, for example, when a

document is transferred to an agent who has not specified a need for the document or a need is specified but no document is transferred. Automatic detection of such errors guarantees a level of formal system description verification not presently available to the auditor.

The third component is a query system which allows the auditor to study corporate-wide control issues. The computer, once equipped with a complete model of the system, can answer questions posed by the auditor about the behavior and characteristics of the total system in question. For example, computer-implementable algorithms can be designed that will analyze the system description and establish that when a document is sent, either an agent receives it or it is deposited in a file. Furthermore, it can establish whether documents are to be destroyed, whether desirable comparisons are to be executed, and under what conditions these comparisons can be subverted. Thus, a wide range of internal control and security issues can be considered by the machine-based analysis process, thereby assisting in validating the procedures the firm asserts that it uses. This vastly expands the utility of the system from simply a documentation procedure to a viable audit tool. Auditors can use it to obtain answers to specific questions about the internal control system and can use the system in the same way they might use their own analytic abilities but enhanced by computer speed and precision. As the scope of the auditor's problem grows, such automated analysis will become cost-beneficial.

Finally, a monitor capable of evaluating and directing an operating office on a real-time basis will be possible. Such a monitor would permit the interface of a control evaluation system and actual events in the office. The result will be a virtually continuous audit. Such an operating system would employ computer audit system actions as specified by the system description on a continuous basis as a means of enforcing office activity compliance. An interface of this nature may well be critical to the success of automated office information systems. In this context, TICOM II may be viewed as the macro control oriented description of such a system. As such it is a design and control tool.

We will present the first component of TICOM II by means of an example based on a cash disbursements system. The next section will present the cash disbursements system flowchart, the language commands

necessary for system description purposes, and the resulting computer compatible system description. This is followed by a somewhat more complete description of the characteristics of the language employed. Last, we discuss the analytic procedures employed by the model in responding to the auditor queries.

#### TICOM II Commands and Cash Disbursements System

One objective of our internal control description language is to capture all of the information represented by the flowchart approach and processed by auditors. We wish to do this with as little loss of generality as possible. The essential characteristics of an internal control system can be imputed from graphical models such as that of figure 1. It is clear that files, documents, document transfers, comparisons, and separation of duties are important internal control features.

The primary aim of this modeling process, as discussed, is the computer analysis of the internal control system. A first approach presents a language that resembles a computer language. Future research may permit the use of even more natural language representations.

A model represented in TICOM II captures both control and data flow information as well as the static functionality and capabilities of individual organizational units. This is achieved by describing the individual activities in terms of the fundamental operations performed on the system documents.

The commands listed in figure 2 represent those we believe to be sufficient to describe all aspects of internal control systems currently accommodated by flowchart and narrative means. We do not propose that the list is complete; however, the commands list can be extended at only moderate cost. In addition to these basic commands, the basic elements of the language are objects, processors, and a transactions concept. Figure 3 presents the resulting internal control system description based on the TICOM II language. The following sections of this paper explain and elaborate on the language employed in figure 3. Figure 4 provides an example mapping between the flowchart of figure 1 and the commands and description in figures 2 and 3.

## Discussion of the Cash Disbursements Model

The cash disbursements model is a transaction-based system driven by the arrival of the documentation for a purchase. Accounts payable CLERK2 initiates action by fetching the purchase documentation from the UNPAID-VOUCHER-FILE. This documentation is prepared and inserted into the UNPAID-VOUCHER-FILE by the purchasing system, which is not shown. In this way, the UNPAID-VOUCHER-FILE is the interface between the purchasing system and the cash disbursements system. The purchase documentation is composed of two voucher copies and copies of the purchase order, receiving report, requisition, and invoice. CLERK2 then performs a series of comparisons to validate the proper recording and processing of the transaction. The documents are compared for corresponding vendors, product descriptions, product numbers, quantities, and price. The failure of any one of these tests initiates a REVIEW. (An expanded version of this example would include a specific description of the review process and its impact on the control system.) Once the purchase documentation is verified, it is transferred to the DEPT-HEAD of Accounts Payable for authorization if the purchase amount, VOUCHER1.AMOUNT, exceeds \$9,999.99. DEPT-HEAD then authorizes or blocks the transaction by setting the APPROVAL field of the vouchers to OK or NOTOK respectively. In this case, CLERK2 awaits the return of the documentation and verifies that the document is properly authorized. If an authorization is not indicated, another REVIEW would be initiated. All verified documents, except for copy two of the voucher, are transferred to CLERK1 of CASH-DISBURSEMENTS. VOUCHER2 is sent to CLERK3 of ACCOUNT-PAYABLE.

CLERK1 of CASH-DISBURSEMENTS is shown WAITING for the forms passed to him/her by CLERK2 of ACCOUNT-PAYABLE. CLERK1 then performs a series of tests similar to the ones performed by CLERK2 to verify the documentation. If the documentation passes these control checks, CLERK1 cancels each document by ASSIGNING the state CANCELLED to the CANCELLATION-FIELD which was previously in an unCANCELLED state. CLERK1 then transfers the first voucher copy to CLERK2 of CASH-DISBURSEMENTS and finally PUTS (stores) the rest of the documents in the PURCHASE-HISTORY-FILE. Once the transfer of VOUCHER1 is made to CLERK2 of CASH-DISBURSEMENTS, CLERK2 is free to proceed.

At about the same instant that CLERK1 of CASH-DISBURSEMENTS received the required documentation, CLERK3 of ACCOUNT-PAYABLE could have received VOUCHER2. Upon receipt of VOUCHER2, CLERK3 posts an entry to the VOUCHER-REGISTER and TRANSFERS VOUCHER2 to CLERK1 of PURCHASE-EXPENSE-LEDGER. Since CLERK1 of CASH-DISBURSEMENTS and CLERK3 of ACCOUNT-PAYABLE are independent tasks initiated almost simultaneously, they correspond to tasks that can be performed concurrently.

The processing continues with the processing of one task initiating the startup of others until a REVIEW traps the transaction or processing completes normally with all purchase documentation verified and archived, registers and ledgers balanced, and canceled checks and deposit slips returned from the BANK.

Figure 4 shows the GENERAL-LEDGER task in flowchart form and the corresponding TICOM II commands. The task begins upon receipt of the VOUCHER-REGISTER, CHECK-REGISTER, and PURCHASE-EXPENSE-LEDGER tapes. Next, the CHECK-REGISTER and the VOUCHER-REGISTER tapes are checked to see if they balance. If not, a REVIEW is initiated; otherwise an entry is posted to the GENERAL-LEDGER. This is done by first retrieving a blank GENERAL-LEDGER-ENTRY from a repository initialized with such blanks; second, completing the account number and description, date, amount and type (debit-credit) fields of the GENERAL-LEDGER-ENTRY; and third, PUTting the GENERAL-LEDGER-ENTRY into the GENERAL-LEDGER. Finally the tapes are archived in the TAPE-HISTORY-FILE.

#### Internal Control Language Concepts

We now present a brief overview of some of the fundamental aspects of the language. The most basic elements of the language are objects, processors, repositories, standard data types, and a transaction concept.

System objects are items that are processed and managed by the system; they may be forms, checks, cash, goods, and data records. Each object is labeled by a unique identifier which is assigned by the user. An identifier is any string of characters; A-Z, 0-9, and '-' (hyphen), such that the first character of the string is a letter of the alphabet and the last character is not a '-' (hyphen). An object is defined by its attributes in terms of data types, Integer, Real, Character,



Boolean, and Scalar. For example, cash may be defined as an object with attribute amount which is of type real, a dollar and cent figure. A check may be defined as an object with attributes amount of type real, check number of type integer, and paid-to of type character. A requisition form with an authorization attribute may be defined as type boolean implying this attribute may be in one of three states: unassigned, true, or false.

Processors are also labeled by identifiers and represent both work-related positions and individual employees. Associated with each processor are tasks which represent the functionality and capabilities of the processors to manage and process the system resources. The TICOM II task description is concerned with what portions of the system a processor manages, processes, and controls rather than how the processor performs the duties.

Repositories, also labeled by unique identifiers, serve as storage mediums for objects. The repositories may represent computer data bases, vaults, physical filing systems, or even a desk drawer. Repositories are characterized by the types of objects they are capable of containing and the objects possessed at the start of a transaction.

The system description in TICOM II is transaction-based. A transaction is defined as the actions required to move a set of object(s) from their initial repositories to their destination repositories, to their destruction, or to their detection as being in an error state. The system description is focused on describing the processing required to record any single transaction. Knowledge of how single transactions are processed is relatable to situations concerning concurrent processing of similar transactions.

The TICOM II language is free-formatted; thus indentation and spacing may be preferentially done.

#### Internal Control Description Language Examples

A TICOM II description of a system is composed of five sections: object type definitions, object definitions, repository definitions, system operations, and task assignments. The first, object type definitions, incorporates the concept of elementary abstract data types within a hierarchical classification scheme. The usage of elementary

abstract data types in programming languages such as PASCAL [9] has enhanced human comprehension of algorithms that are "reasonably" coded and thus serves as the primary incentive for including elementary abstract data typing in TICOM II. Declarations of objects that are to be managed and processed by the system are contained in the object definitions section. The repository definition section is used to specify sources and destinations for objects. The system operations section provides the means for describing the control and data flows and also the functionality of work-related positions. The task assignment section is used to assign individual employees to particular positions.

#### Definition and Typing of System Objects

Before presentation of the system description, TICOM II is constructed to possess little prior built-in knowledge of the objects managed by internal control systems. The system has no preconception of documents, cash, or inventory items. Such knowledge could facilitate the object definition process for objects recognizable by TICOM II. However, if object definitions were restricted to those anticipated by the designers of TICOM II, only a limited class of internal control systems could be represented by the language. In order to permit the description of a wide range of internal control systems, TICOM II is developed with an abstract mechanism for defining objects. This mechanism is presented here.

An object is a labeled item with associated attributes. The labeled item is a proxy for an instance of the specified object. The object and references to it are in themselves abstractions, since an instance of the object is never examined. Objects are distinguished from one another by their object types, which are labels that specify the attributes that are to be associated with objects of said type. An attribute is a labeled characteristic which necessarily belongs to the object. A characteristic is referenced by specifying its label. Attributes are distinguished from one another by their attribute types. An attribute type specifies the data type that can be employed to represent an instance of an attribute. An attribute type labels the "nature of the information" that an attribute conveys. Object classifications are labels that represent a collection of objects. The

objects to be included in the classification are specified by listing the object types of the desired objects and/or listing previously defined object classifications. Objects of the specified object types and/or previously defined classifications become members of the newly formed object classification.

The relationships between objects, object types, attributes, attribute types, data types, and object classifications are summarized in figure 5. Boxes are labeled to denote the definition components. A directed arc signifies the composition of the definition component from which the arc emanates. The 1:N labeling of an arc represents a one-to-many relationship. Conversely, N:1 represents a many-to-one relationship. Thus the diagram is interpreted as follows.

An object classification is defined in terms of other object classifications and/or object types. The 1:N and the N:1 relationships are inferred by the plurality and singularity of the definition components. Each object type belongs to one or more object classifications and has one or more objects defined as being of its type. An object may be declared to be of only one object type and inherits the attributes associated with its object type. An object type is composed of one or more attributes. Each attribute has an attribute type. Any given attribute type may be associated with more than one attribute. Similarly, each attribute type is associated with a particular data type, and any given data type may be associated with more than one attribute type. Figure 6 is a partial description of the knowledge base concerning system objects that TICOM II would generate from the OBJECT TYPE DEFINITIONS and the OBJECT DEFINITIONS sections of the cash disbursements example. The following paragraph relates figure 6 to the cash disbursements example and discusses the need for, and advantages of, such an elaborate definition scheme.

Listed below are the statements taken from the cash disbursements example that are used to define the object type CHECKS.

Example: NAME = CHARACTER.  
 DATE = CHARACTER.  
 DOLLAR = REAL.  
 CANCELLATION-STAMP = (CANCELLED).

CHECKS = PAYEE:NAME; DATE-PAID, DATE-CASHED:DATE;  
 AUTHORIZATION:NAME; AMOUNT-PAID:DOLLAR;  
 CANCELLATION-FIELD:CANCELLATION-STAMP.

CHECK:CHECKS.

The last statement listed is taken from the OBJECT DEFINITION section and thus declares that CHECK is an object of type CHECKS. Thus, CHECKS is an object type and the object CHECK inherits the attributes associated with object type CHECKS. The rest of the statements of this example are taken from the OBJECT TYPE DEFINITIONS section, thereby remaining consistent with the fact that CHECKS is an object type. The attributes of CHECKS are the labels listed under the definition of CHECKS, namely, PAYEE, DATE-PAID, DATE-CASHED, AUTHORIZATION, AMOUNT-PAID, and CANCELLATION-FIELD. Each attribute is associated with an attribute type. For instance, PAYEE is of attribute type NAME. NAME is defined by the first line of the example in terms of the type of data that is to be used to represent an instance of the attribute. By definition, GEORGE E. SMITH is CHARACTER data and would be a valid instance of PAYEE. Likewise, attributes DATE-PAID and DATE-CASHED are of attribute type DATE which is also of data type CHARACTER. Despite the fact that the instances of DATE-PAID and PAYEE are represented with CHARACTER data, DATE-PAID and PAYEE are defined as attributes carrying different natures of information. This differentiation between the two attributes is known since the two attributes have different attribute types. Therefore any internal checks (comparisons) or operations between objects or their attributes that differ in type is flagged as being in error by TICOM II.

The CANCELLATION-FIELD attribute of CHECKS is stated to be of attribute type CANCELLATION-STAMP. CANCELLATION-STAMP is a scalar data type. The permissible states of this scalar data type are listed within the parentheses (see example). In this particular case, the only permissible states for an attribute of type CANCELLATION-STAMP is CANCELLED and unassigned (blank).

The need for such elaborate typing is evident if one only considers the opposite extremes, i.e., a very simple typing scheme. In such a scheme all objects would be of type CHARACTER. Object attributes would be of type Character, Boolean, Real, and Integer. Any two objects or attributes can be compared or operated on if they have matching types. So virtually any two objects or attributes could be compared or copied to one another despite their empirical differences. Attributes would only be distinguished by name, not by the nature of the information they convey. An internal control analysis system that confused dates with names and checks with receiving reports would have little worth.

In TICOM II, the level of object definition and typing is left to the judgment of the designer. For example, if the system designer did not want to consider a low level of detail for objects, he/she could have defined the object CHECK with the following two declarations.

```
EXAMPLE: CHECKS = CHARACTER.  
CHECK: CHECKS;
```

Obviously, no references could be made to the PAYEE, AMOUNT-PAID, etc., but the object CHECK could still be referenced with the TICOM II commands in the usual way. However, the absence of detail would make it impossible to determine who signs the check and who writes the check if these tasks are split between employees. It would also be impossible to describe adequately the check reconciliation process since CHECK would not have a check number, payee, date paid, and amount paid. However, it would be possible to describe the transfer of the check from the bank to payroll accounting, which in turn could place the check into a reconciled check file. Implicit with the filing of the check would be the check reconciliation process. Once the system is described at the higher level, further detail can be added, resulting in a top-down design process.

As mentioned, another purpose of the OBJECT TYPE DEFINITIONS section is to classify objects. This is accomplished by nesting the object type definitions within a hierarchical classification schema. In the cash disbursement example, CHECKS are shown to be classified as a PHYSICAL-ASSET and also as an ASSET, since CHECKS' declaration is positioned within the scope of the PHYSICAL-ASSETS and ASSETS headers.

The specification of hierarchical classifications of object types and attribute types is useful for the query analysis. Once the classifications are specified, questions can be posed utilizing the classification schema. For instance, the question, "Who can access INDIRECT-ASSETS?" can now be interpreted by the computer. That is, objects of type REQUISITIONS, RECEIVING-REPORTS, INVOICES, PURCHASES-ORDERS, VOUCHERS-T1, and VOUCHERS-T2 are classified as INDIRECT-ASSETS. Access to an object of one of these types is access to an INDIRECT-ASSET.

#### Repository Definitions

Repositories serve as sources of objects, destinations of objects, and temporary storage of objects. A repository is characterized by its label, the type of objects it may contain, and its initial state, for example:

BLANK-CHECK-SUPPLY CONTAINS CHECKS INITIALIZED WITH CHECK

states that repository BLANK-CHECK-SUPPLY is designed to hold objects of type CHECKS and that at the start of a transaction BLANKCHECK-SUPPLY contains object CHECK, a system resource made available to the transaction. Repositories which are not initialized are assumed to be empty at the start of the transaction.

As in the OBJECT TYPE DEFINITIONS section, repositories in the REPOSITORY DEFINITIONS section may be hierarchically classified. In the cash disbursement example, all of the repositories comprise a single class, PHYSICAL-LOCATIONS.

#### System Operations

The SYSTEM OPERATIONS section is used to describe procedurally the functions and capabilities of work-related positions, departments, and decisions in regard to the processing of a transaction. The associations of work-related positions, departments, and divisions are specified by the same hierarchical scheme mentioned. For example, in the cash disbursements system, CLERK1 and CLERK2 are work-related positions within department CASH-DISBURSEMENTS. Departments

ACCOUNT-PAYABLE and CASH-DISBURSEMENTS are shown as being hierarchically unrelated and distinct. The functions and capabilities of a work-related position are those listed within the scope of the position identifier. The collective composition of functions and capabilities of work-related positions associated with the same classification (i.e., department) comprise the functions and capabilities of that classification or grouping. Thus, the functions and capabilities granted to the ACCOUNT-PAYABLE department are the sum total of the functions and capabilities assigned to CLERK2, CLERK3, and DEPT-HEAD. The hierarchical nesting of positions, departments, and divisions and its depth are arbitrarily dependent upon the system designer. Again, these classifications are defined to facilitate the phrasing of questions concerning the behavior of the system.

The functions and capabilities of a work-related position are denoted by listing under each position a set of tasks to be performed by an employee serving within the capacity of that position. A task is defined to be a logical contiguous set of instructions that process and manage system objects and is to be performed in that order indicated. The individual carrying out the task is termed processor or agent.

#### Error Processing Examples

The three examples following demonstrate the kind of errors and inconsistencies that can be detected by the analysis component of TICOM II.

EXAMPLE: GET CHECK FROM BLANK-CHECK-SUPPLY.

This statement is valid if the BLANK-CHECK-SUPPLY contains the system object CHECK at the time of its execution. The result of the command is that the CHECK is removed from the BLANK-CHECK-SUPPLY repository and is made available to the requesting agent for processing.

EXAMPLE: TRANSFER CHECK TO VENDOR.

This statement is valid if the initiating agent possesses CHECK at the time of its execution and if agent VENDOR has a matching WAIT FOR CHECK

instruction to denote reception of the CHECK. The result of the command is that the CHECK leaves the possession of the initiating agent to become a possession of VENDOR.

```
EXAMPLE: IF CHECK-REGISTER-TAPE.TOTAL < > VOUCHER-REGISTER-TAPE.
          TOTAL THEN
            REVIEW;
          END TASK;
        END IF;
```

The statement is valid if the fields being compared are of the same data type. In this case both fields are of type DOLLAR and thus the comparison is legitimate. If the result of the not-equal comparison is true, then the statement REVIEW is to be performed next. Otherwise, the REVIEW statement is bypassed, and the first instruction following the END IF statement is executed.

#### Task Assignments

The TASK ASSIGNMENTS section is used to assign individual employees to particular positions. An individual employee is designated by a unique identifier, in the case of the cash disbursements example, by an identification number. An individual may be assigned to more than one position indicating a split appointment.

#### EXAMPLE:

```
ACCOUNT-PAYABLE.CLERK2 IS 839,963.
```

This indicates that employees 839 and 963 are both CLERK2s in the ACCOUNT-PAYABLE department authorized to perform the functions designated for that position.

#### Synopsis of the TICOM II Model

As demonstrated, the TICOM II language is an effective medium for representing a system in a formalized manner. In TICOM II a system is formalized as

- (1) a set of object type definitions,
- (2) a set of object type classifications,



- (3) a set of objects,
- (4) a function mapping objects to object classifications,
- (5) a set of repositories,
- (6) a function mapping objects to repositories,
- (7) a set of repository classifications,
- (8) a function mapping repositories to repository classifications,
- (9) a set of processors (positional),
- (10) a set of commands defined over the objects, repositories, and processors,
- (11) a function mapping processors to commands,
- (12) a set of processor classifications,
- (13) a function mapping processors to processor classifications,
- (14) a function mapping the state of the system from one state to the next in accordance with the specified precedence of operations,
- (15) a set of employees,
- (16) a function mapping the employees to processors (positional),
- (17) an initial state of the system.

The state of the system is described by the collective states of the objects. The state of an object is depicted by the domain in which it resides and the state of each of its components. The state of a component is either unassigned or assigned to a permissible state as determined by the component's type.

The preceding introductory explanation of the TICOM II language concerned itself with showing the general structure of the language and demonstrating the versatility of the language for defining and managing system resources. The next and last section is a brief introduction into the query-processing component of TICOM II.

#### Query System

The major value of the TICOM II system is its ability to automatically analyze properties of the internal control model. Questions are posed in a query language which is expressed in a subject-predicate calculus-like format. Objects and agents defined in the system description are valid subjects in the query language. The grouping of

objects and agents into useful categories in the system description define additional predicates in the query language. To illustrate the query language and its relationship with the system description, consider the following question concerning the cash disbursement system: Who in the accounts payable department authorizes payments? In the query language this question would be phrased as

(IS THERE) [ASSIGN APPROVAL OF v BY p and VOUCHERS(v) and ACCOUNT-PAYABLE (p)]?

To answer the question, the query processor needs to find all possible substitutions for variables v and p that satisfy the question. From the object type definition section of the system description, the query processor identifies objects of type VOUCHERS-T1 and VOUCHERS-T2 as belonging to the grouping (set) VOUCHERS. In the object definition section VOUCHER1 and VOUCHER2 are declared as being of type VOUCHERS-T1 and VOUCHERS-T2, respectively. In this manner, the query processor associates VOUCHER1 and VOUCHER2 as elements of the set VOUCHERS. The interpretation of VOUCHERS(v) is simply that v is an object which belongs to the set VOUCHERS. Thus, VOUCHERS(v) is a predicate for determining group membership. For VOUCHERS(v) to be true, v must be equated with either VOUCHER1 or VOUCHER2. Similarly, ACCOUNT-PAYABLE(p) is true if p is equated with one of the three positions defined within ACCOUNT-PAYABLE: CLERK2, CLERK3, and DEPT-HEAD.

The first part of the query, ASSIGN APPROVAL OF v BY p, instructs the query processor to examine all ASSIGN statements in the system description for those which initialize the APPROVAL attribute of object v by agent p subject to the constraints:  $v \in \text{VOUCHERS}$  and  $p \in \text{ACCOUNT-PAYABLE}$ . In so doing, the query processor finds that for  $p = \text{DEPT-HEAD}$  and  $v = \text{VOUCHER1}$  or for  $p = \text{DEPT-HEAD}$  and  $v = \text{VOUCHER2}$ , the query is satisfied.

By posing other types of questions that exploit the system representation we have chosen, a wide range of internal control and security issues can be mechanically analyzed, assisting in validating the procedures the firm uses. Typical issues that can be analyzed in this way include (1) the existence or lack thereof of the segregation of

duties at both the aggregate and individual levels; (2) the order of access and processing of documents; (3) the existence of particular processing steps at any given stage of system operation; (4) comparison of job description with actual duties; (5) the existence of necessary conditions for internal checks and comparisons deemed valuable and described by auditors; and (6) the necessary and sufficient conditions for a given set of actions to occur. As the scope of the auditor's problem grows, such automated analysis will prove necessary.

#### Analytic Capabilities of TICOM II

Determination of the necessary and sufficient conditions for a given set of actions to occur is the principal analytic process employed by TICOM II to answer queries concerning state achievability. That is, is it possible for the firm, given its internal control system, to reach a particular state that would be considered undesirable from an internal control perspective?. If it is possible to enter such a state, TICOM II establishes the necessary conditions that will lead to that event. An example of such an event is the creation of a purchase order without the requisition's first being approved. Questions of state achievability are equivalent to questions that ask whether or not it is possible to find a sequence of instructions that leads to the given state, subject to the processing constraints imposed by the internal control system.

In order to illustrate the analytic processes utilized by TICOM II, it is necessary to describe briefly a representation for modeling the operations of an internal control system. Conceptually, the internal control operations can be depicted by a precedence graph. Each node of the graph corresponds uniquely to an instruction in the system description and vice versa. The nodes are interconnected with precedence arcs which specify the order in which the instructions are to be performed. Precedence arcs are labeled to indicate the truth of conditional branches. The conditional branches relate directly to the IF-statements in the system description. A more detailed explanation of the precedence model concept is given by Ellis and Morris (1979).

The first analytic process employed in TICOM II is called a contraction. A contraction is a procedure for eliminating a command from the system without destroying any of the relative properties of the

other commands. In order to establish the relationship of some set of commands, all of the other commands in the system may be contracted out. Intuitively, contraction is performed by removing a node from the precedence graph by combinatorally connecting the input arcs of the node with the output arcs of the node. Figures 7a and 7b show two examples of a precedence graph before and after node B is contracted. Figure 7a stipulates that node B can begin processing after the termination of processing in node A. Similarly node C follows node B. By contracting out node B, we preserve the implied relationship between nodes A and C, namely that node C proceeds after termination of node A. Figure 7b stipulates that node B can proceed after the termination of node A and that node C and/or node D can proceed after node B terminates. Obviously, the graph resulting from the contraction of node B preserves the precedence relationships between nodes A, C, and D. For cases in which the input and/or output arcs of the node are labeled with conditional assertions, the appropriate assertions for the paired input-output arcs are combined.

In addition to contraction, TICOM II possesses simplification rules for reducing the complexity of precedence graphs. Due to space limitations, it is not practical to present all the simplification rules used in TICOM II. Rule 1 is a special case of a more general rule for simplifying cycle implication. Figure 7c exhibits simplification of Rule 1, which states that an arc which loops back to the node from which it started may be removed if the conditional assertions ( $\sim r_1$ ) for making the cycle are the negation of the conditional assertions ( $r_1$ ) for leaving the cycle. The soundness of the rule is obvious. The subsequent node of A can only proceed if node A ends with  $r_1$  being asserted, else node A cycles. After termination of node A, either node A or its immediate successor may proceed but not both, since that would assert  $r_1 \wedge \sim r_1$ , a contradiction. Therefore, node A's successor may proceed if and only if node A terminates with the conditional assertion  $r_1$ .

Figure 7d exhibits simplification Rule 2, which states that if node B follows node A with  $r_1 \wedge r_2$  or  $\sim r_1 \wedge r_2$  being asserted, then it logically follows that node B follows node A with  $r_2$  being asserted, where  $r_1$  and  $r_2$  denote a collection of conditional assertions.

Intuitively, the general procedure for establishing the relationships between one or more nodes of the model is that (1) the exact nodes under study are marked; (2) repeatedly an unmarked node is selected and contracted out of the model until only marked nodes remain; and (3) the simplification rules are used to simplify the precedence constraints. Figures 8 through 15 show a step-by-step illustration of the analysis process just described. Node 6 is the marked node under question. Figure 15 shows the final result of the process, the necessary and sufficient condition for reaching node 6, i.e., entering a given state, is assertion  $T_1$ . It is left to the reader to validate the example.

#### Comments on the Empirical Appeal of Contraction and Simplification

As noted, there are two approaches concerning the audit techniques associated with internal control. The cycles approach isolates the related tasks of a given cycle from tasks which are not an integral part of the cycle. These tasks which are external to the cycle are grouped into one or more "black boxes" that feed inputs to the cycle and receive the outputs of the cycle. By so doing, the auditor reduces the complexity of the internal control system to a more manageable size and yet maintains a system-wide perspective. The second approach, a transactions and account classification approach, also isolates related tasks but by different criteria. Analysis is again conducted on a reduced system.

In both cases, the total system is reduced and simplified to a collection of related tasks. Within each collection of tasks, detailed analysis of the internal control mechanisms are made. Contraction and simplification are analogous steps within TICOM II and are consistent with present auditing practices. TICOM II maintains a cycle perspective by marking only tasks (nodes) related to the cycle. The unmarked nodes are then contracted out, and the precedence constraints associated with inputs to the cycle are simplified to contain only the necessary conditions (conditional assertions) for the inputs to enter the cycle. The model of the isolated cycle can then be further analyzed with TICOM II while maintaining a system-wide perspective. An analogous argument can be made for the transactions and account classification approach.

The conclusion here is that once a total system-wide description of the firm's internal controls is done in TICOM II, it is possible for the auditor or system designer to reduce and simplify the model to any specified cycle, transaction, account classification, or other classification of tasks and still maintain a system-wide perspective simply by specifying the task selection criteria for TICOM II implementation.

#### Conclusions

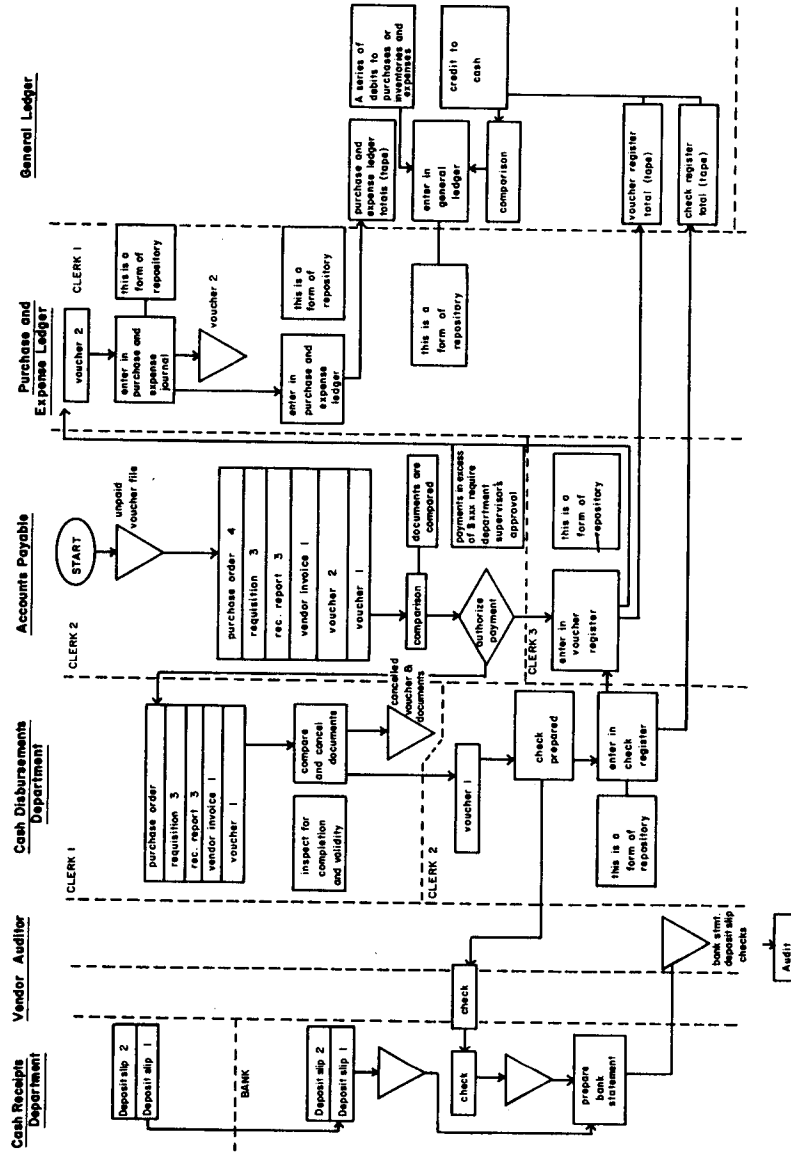
We believe that TICOM II will be a valuable tool in the description, design, analysis, and evaluation of internal control systems. Accountants and auditors will find that, with the aid of the computer, a more professionally satisfactory job can be done. Further, TICOM II may be viewed as a macro design language in an Automated Office Information System (OIS) setting. The importance of this is that TICOM II is uniquely suitable from the viewpoint of accounting and auditing control. Much of the existing OIS literature has concentrated on efficiency, often to the point of creating an efficient but uncontrolled system.

We are still in the process of refining the TICOM II language and in determining the necessary software to support the processes described.

#### Authors' Note

This project was funded in part by a grant from the Peat, Marwick, Mitchell Foundation through its Research Opportunities in Auditing Program. The views expressed herein are those of the authors and do not necessarily reflect the views of the Peat, Marwick, Mitchell Foundation. We gratefully acknowledge also the fellowship support made possible by a grant from IBM.

Figure 1. Cash Disbursements System



COMMAND	DESCRIPTION
ASSIGN attribute-list OF object;	Specifies which attributes of a particular object are to be assigned values.
MODIFY attribute-list OF object;	Specifies which previously assigned attributes of a particular object are to be reassigned values.
DESTROY object-list;	Specifies which objects are to be destroyed.
IF boolean-expression THEN true instructions ELSE false instructions END IF;	Specifies a boolean expression, a simple or compound comparison, whose truth determines which disjoint set of instructions are to be performed next.
IF boolean-expression THEN true instructions END IF;	
TRANSFER object-list TO agent;	Specifies that the data objects listed are to be transferred to another agent or organizational unit.
WAIT FOR object-list;	Specifies that the agents' processing is blocked until he receives the objects listed.
PUT object-list INTO repository;	Specifies that the objects listed are to be placed into the designated repository.
GET object-list FROM repository;	Specifies that the objects listed are to be retrieved from the designated repository.
COPY target-object GIVING duplicate object-list;	Specifies that the target object is to be copied creating the designated duplicate objects.
END TASK;	Specifies the end of a task.
REVIEW	Signifies the entrapment of an error or a discrepancy.

Figure 2 The Commands



## SYSTEM:

OBJECT TYPE DEFINITIONS:  
 PART-NO = CHARACTER.  
 BANK-ACCT-NO = CHARACTER.  
 DATE = CHARACTER.  
 NAME = CHARACTER.

DOLLAR = REAL.  
 CANCELLATION-STAMP = (CANCELLED).  
 ACCT-TITLE = CHARACTER.  
 ACCT-NO = CHARACTER.  
 VOUCHER-NO = CHARACTER.  
 CHECK-NO = CHARACTER.  
 DEPT-TITLE = CHARACTER.  
 PRODUCT-TITLE = CHARACTER.  
 VENDOR-TITLE = CHARACTER.  
 DISCOUNT-DESC = CHARACTER.  
 COUNT = INTEGER.  
 APPROVAL-STAMP = (OK,NOTOK).  
 DEPT-SUPER-APPROVAL-STAMP = (WJG,TSE).  
 DEBIT-CREDIT = (DEBIT,CREDIT).

## ASSETS:

PHYSICAL-ASSETS:  
 CHECKS = PAYEE:NAME; DATE-PAID, DATE-CASHED:DATE; AUTHORIZATION:NAME  
 AMOUNT-PAID:DOLLAR; CANCELLATION-FIELD:CANCELLATION-STAMP.  
 END PHYSICAL-ASSETS.

## INDIRECT-ASSETS:

REQUISITIONS = SOURCE-DEPT:DEPT-TITLE; SOURCE-INDIVIDUAL:NAME;  
 DATE-SUBMITTED:DATE; PRODUCT-DESCS:PRODUCT-TITLE;  
 PRODUCT-NUMBERS:PART-NO; QTY:COUNT;  
 APPROVAL:DEPT-SUPER-APPR-STAMP;  
 CANCELLATION-FIELD:CANCELLATION-STAMP.

RECEIVING-REPORTS = VENDOR:VENDOR-TITLE; DATE-RECEIVED:DATE;  
 PRODUCT-DESCS:PRODUCT-TITLE;  
 PRODUCT-NUMBERS:PART-NO; QTY:COUNT;  
 APPROVAL:APPROVAL-STAMP;  
 CANCELLATION-FIELD:CANCELLATION-STAMP.

PURCHASE-ORDERS = VENDOR:VENDOR-TITLE; DATE-PURCHASED:DATE;  
 PRODUCT-DESCS:PRODUCT-TITLE;  
 PRODUCT-NUMBERS:PART-NO; PRICE:DOLLAR;  
 CANCELLATION-FIELD:CANCELLATION-STAMP;  
 QTY:COUNT; APPROVAL:APPROVAL-STAMP.

INVOICES = VENDOR:VENDOR-TITLE; DATE-SHIPPEE:DATE;  
 PRODUCT-DESCS:PRODUCT-TITLE; PRODUCT-NUMBERS:PART-NO;  
 PRICE:DOLLAR; QTY:COUNT; DISC-TERMS:DISCOUNT-DESC;  
 CANCELLATION-FIELD:CANCELLATION-STAMP.

## VOUCHERS:

VOUCHERS-T1 = VENDOR:VENDOR-TITLE; DATE-PAID:DATE;  
 ACCOUNT-NUMBER:ACCT-NO; AMOUNT:DOLLAR;  
 ACCOUNT-DESC:ACCT-TITLE;  
 APPROVAL:APPROVAL-STAMP;  
 CANCELLATION-FIELD:CANCELLATION-STAMP.

Figure 3

VOUCHERS-T2 = VENDOR:VENDOR-TITLE; DATE-PAID:DATE;  
 ACCOUNT-NUMBER:ACCT-NO; AMOUNT:DOLLAR;  
 ACCOUNT-DESC:ACCT-TITLE;  
 APPROVAL:APPROVAL-STAMP.

END VOUCHERS.  
 END INDIRECT-ASSETS.  
 END ASSETS.

DEPOSIT-SLIPS = ACCOUNT-NUMBER:BANK-ACCT-NO.  
 DATE-OF-DEPOSIT:DATE; DEPOSITOR:NAME; AMOUNT:DOLLAR.

ACCT-PAID-NOTIFICATIONS = VOUCHER-NUMBER:VOUCHER-NO; CHECK-NUMBER:CHECK-NO.

CHECK-REGISTER-ENTRIES = CHECK-NUMBER:CHECK-NO; PAYEE:NAME; ENTRY-DATE:DATE;  
 AMOUNT:DOLLAR; TYPE:DEBIT-CREDIT;  
 VOUCHER-NUMBER:VOUCHER-NO.

VOUCHER-REGISTER-ENTRIES = PAYEE:NAME; ENTRY-DATE:DATE;  
 VOUCHER-NUMBER:VOUCHER-NO;  
 AMOUNT:DOLLAR; CHECK-NUMBER:CHECK-NO.

BANK-STATEMENTS = CHECK-NUMBER:CHECK-NO; DATE-CASHED:DATE; CHECK-AMT:DOLLAR;  
 DEPOSIT-NUMBER:BANK-ACCOUNT-NUMBER;  
 DATE-OF-DEPOSIT:DATE; DEPOSIT-AMT:DOLLAR.

ACCOUNTING-ENTRIES = ENTRY-DATE:DATE; ACCOUNT-NUMBER:ACCT-NO;  
 ACCOUNT-DESC:ACCT-TITLE; AMOUNT:DOLLAR;  
 TYPE:DEBIT-CREDIT.

TOTAL-FIGURES = ENTRY-DATE:DATE; TOTAL:DOLLAR.

NET-FIGURES = ENTRY-DATE:DATE; NET-TOTAL,GRAND-NET-TOTAL:DOLLAR;  
 ACCOUNT-NUMBER:ACCT-NO.

END OBJECT TYPE DEFINITIONS.

OBJECT DEFINITIONS:

CHECK-REGISTER-TOTAL,VOUCHER-REGISTER-TOTAL,VOUCHER-REGISTER-TAPE,  
 CHECK-REGISTER-TAPE:TOTAL-FIGURES;  
 PURCHASE-EXPENSE-LEDGER-TOTAL,PURCHASE-EXPENSE-LEDGER-TAPE:NET-FIGURES;  
 GENERAL-LEDGER-ENTRY:ACCOUNTING-ENTRIES;  
 BANK-STATEMENT:BANK-STATEMENTS;  
 PURCHASE-EXPENSE-LEDGER-ENTRY:ACCOUNTING-ENTRIES;  
 PURCHASE-EXPENSE-JOURNAL-ENTRY:ACCOUNTING-ENTRIES;  
 REQ3:REQUISITIONS;  
 PURCHASE-ORDER4:PURCHASE-ORDERS;  
 RR3:RECEIVING-REPORTS;  
 INVOICE1:INVOICES;  
 VOUCHER1:VOUCHERS-T1;  
 VOUCHER2:VOUCHERS-T2;  
 CHECK:CHECKS;  
 CHECK-REGISTER-ENTRY:CHECK-REGISTER-ENTRIES;  
 VOUCHER-REGISTER-ENTRY:VOUCHER-REGISTER-ENTRIES;  
 DEPOSIT-SLIP1,DEPOSIT-SLIP2:DEPOSIT-SLIPS;  
 ACCT-PAID-NOTIFICATION:ACCT-PAID-NOTIFICATIONS;

END OBJECT DEFINITIONS.

Figure 3 (Cont.)

## REPOSITORY DEFINITIONS:

## PHYSICAL-LOCATIONS:

UNPAID-VOUCHER-FILE CONTAINS VOUCHERS, REQUISITIONS,  
PURCHASE-ORDERS, INVOICES, RECEIVING-REPORTS  
INITIALIZED WITH VOUCHER1, VOUCHER2, PURCHASE-ORDER4,  
RR3, INVOICE1.

CANCELLED-CHECK-FILE CONTAINS CHECKS.

SUPPLY-BLANK-VOUCHER-REGISTER-TAPES CONTAINS TOTAL-FIGURES  
INITIALIZED WITH VOUCHER-REGISTER-TAPE.

SUPPLY-BLANK-CHECK-REGISTER-TAPES CONTAINS TOTAL-FIGURES  
INITIALIZED WITH CHECK-REGISTER-TAPE.

SUPPLY-BLANK-PURCHASE-EXPENSE-LEDGER-TAPE CONTAINS NET-FIGURES  
INITIALIZED WITH PURCHASE-EXPENSE-LEDGER-TAPE.

TAPE-HISTORY-FILE CONTAINS TOTAL-FIGURES, NET-FIGURES.

DEPOSIT-SLIP-FILE CONTAINS DEPOSIT-SLIPS.

SUPPLY-BLANK-NOTIFICATIONS CONTAINS ACCOUNT-PAID-NOTIFICATIONS  
INITIALIZED WITH ACCOUNT-PAID-NOTIFICATION.

SUPPLY-BLANK-DEPOSIT-SLIPS CONTAINS DEPOSIT-SLIPS  
INITIALIZED WITH DEPOSIT-SLIP1.

SUPPLY-BLANK-VOUCHER-REGISTER CONTAINS VOUCHER-REGISTER-ENTRIES  
INITIALIZED WITH VOUCHER-REGISTER-ENTRY.

VOUCHER-REGISTER CONTAINS VOUCHER-REGISTER-ENTRIES,  
TOTAL-FIGURES INITIALIZED WITH VOUCHER-REGISTER-TOTAL.

PURCHASE-HISTORY-FILE CONTAINS VOUCHERS, REQUISITIONS,  
PURCHASE-ORDERS, INVOICES, RECEIVING-REPORTS.

BLANK-CHECK-SUPPLY CONTAINS CHECKS INITIALIZED WITH CHECK.

SUPPLY-BLANK-CHECK-REGISTER-ENTRIES CONTAINS CHECK-REGISTER-ENTRIES  
INITIALIZED WITH CHECK-REGISTER-ENTRY.

CHECK-REGISTER CONTAINS CHECK-REGISTER-ENTRIES, TOTAL-FIGURES  
INITIALIZED WITH CHECK-REGISTER-TOTAL.

GENERAL-LEDGER CONTAINS ACCOUNTING-ENTRIES.

SUPPLY-BLANK-GENERAL-LEDGER-FORMS CONTAINS  
ACCOUNTING-ENTRIES INITIALIZED WITH GENERAL-LEDGER-ENTRY.

PURCHASE-EXPENSE-VOUCHER-FILE CONTAINS VOUCHERS.

BANK-STATEMENT-SUPPLY CONTAINS BANK-STATEMENTS  
INITIALIZED WITH BANK-STATEMENT.

AUDIT-FILE CONTAINS BANK-STATEMENTS, DEPOSIT-SLIPS, CHECKS.

SUPPLY-BLANK-PURCHASE-EXPENSE-JOURNAL-ENTRIES CONTAINS  
ACCOUNTING-ENTRIES INITIALIZED WITH

Figure 3 (Cont.)

PURCHASE-EXPENSE-JOURNAL-ENTRY.  
 SUPPLY-BLANK-PURCHASE-EXPENSE-LEDGER-ENTRIES CONTAINS  
 ACCOUNTING-ENTRIES INITIALIZED WITH  
 PURCHASE-EXPENSE-LEDGER-ENTRY.  
 PURCHASE-EXPENSE-JOURNAL CONTAINS ACCOUNTING-ENTRIES.  
 PURCHASE-EXPENSE-LEDGER CONTAINS ACCOUNTING-ENTRIES,  
 NET-FIGURES INITIALIZED WITH PURCHASE-EXPENSE-LEDGER-TOTAL.  
 END PHYSICAL-LOCATIONS.  
 END REPOSITORY-DEFINITIONS.

## SYSTEM OPERATIONS:

## ACCOUNT-PAYABLE:

```

CLERK2:
  GET VOUCHER1,VOUCHER2,REQ3,PURCHASE-ORDER4,RR3,
  INVOICE1 FROM UNPAID-VOUCHER-FILE:
  IF PURCHASE-ORDER4.VENDOR <> RR3.VENDOR
  OR PURCHASE-ORDER4.VENDOR <> INVOICE1.VENDOR
  OR PURCHASE-ORDER4.VENDOR <> VOUCHER1.VENDOR THEN
  REVIEW:
  END TASK:
  END IF:
  IF PURCHASE-ORDER4.PRODUCT-DESC <> REQ3.PRODUCT-DESC
  OR PURCHASE-ORDER4.PRODUCT-DESC <> RR3.PRODUCT-DESC
  OR PURCHASE-ORDER4.PRODUCT-DESC <> INVOICE1.PRODUCT-DESC THEN
  REVIEW:
  END TASK:
  END IF:
  IF PURCHASE-ORDER4.PRODUCT-NUMBERS <> REQ3.PRODUCT-NUMBERS
  OR PURCHASE-ORDER4.PRODUCT-NUMBERS <> RR3.PRODUCT-NUMBERS
  OR PURCHASE-ORDER4.PRODUCT-NUMBERS <> INVOICE1.PRODUCT-NUMBERS THEN
  REVIEW:
  END TASK:
  END IF:
  IF PURCHASE-ORDER4.PRICE <> INVOICE1.PRICE THEN
  REVIEW:
  END TASK:
  END IF:
  IF PURCHASE-ORDER4.QTY <> REQ3.QTY
  OR PURCHASE-ORDER4.QTY <> RR3.QTY
  OR PURCHASE-ORDER4.QTY <> INVOICE1.QTY THEN
  REVIEW:
  END TASK:
  END IF:
  IF VOUCHER1.AMOUNT <> INVOICE1.QTY * INVOICE1.PRICE THEN
  REVIEW:
  END TASK:
  END IF:
  IF VOUCHER1.AMOUNT > 9999.99 THEN
  TRANSFER VOUCHER1,VOUCHER2,REQ3,PURCHASE-ORDER4,RR3,
  INVOICE1 TO ACCOUNT-PAYABLE.DEPT-HEAD:
  WAIT FOR VOUCHER1,VOUCHER2,REQ3,PURCHASE-ORDER4,RR3,
  INVOICE1:
  IF VOUCHER1.APPROVAL <> OK
  OR VOUCHER2.APPROVAL <> OK THEN
  REVIEW:
  
```

Figure 3 (Cont.)

```

        END TASK;
        END IF;
        END IF;
        TRANSFER VOUCHER1,REQ3,PURCHASE-ORDER4,RR3,INVOICE1
        TO CASH-DISBURSEMENTS.CLERK1;
        TRANSFER VOUCHER2 TO ACCOUNT-PAYABLE.CLERK3;
        END TASK;
    END CLERK2.

    CLERK3:
        WAIT FOR VOUCHER2;
        GET VOUCHER-REGISTER-ENTRY FROM SUPPLY-BLANK-VOUCHER-REGISTER;
        ASSIGN VOUCHER-NUMBER,PAYEE,ENTRY-DATE,AMOUNT,TYPE
        OF VOUCHER-REGISTER-ENTRY;
        PUT VOUCHER-REGISTER-ENTRY INTO VOUCHER-REGISTER;
        GET VOUCHER-REGISTER-TOTAL FROM VOUCHER-REGISTER;
        ASSIGN TOTAL,ENTRY-DATE OF VOUCHER-REGISTER-TOTAL;
        PUT VOUCHER-REGISTER-TOTAL INTO VOUCHER-REGISTER;
        TRANSFER VOUCHER2 TO PURCHASE-EXPENSE-LEDGER.CLERK1;
        END TASK;

        WAIT FOR ACCOUNT-PAID-NOTIFICATION;
        GET VOUCHER-REGISTER-ENTRY FROM VOUCHER-REGISTER;
        ASSIGN CHECK-NUMBER OF VOUCHER-REGISTER-ENTRY;
        PUT VOUCHER-REGISTER-ENTRY INTO VOUCHER-REGISTER;
        DESTROY ACCOUNT-PAID-NOTIFICATION;
        END TASK;

        MONTHLY GET VOUCHER-REGISTER-TOTAL FROM VOUCHER-REGISTER;
        GET VOUCHER-REGISTER-TAPE FROM SUPPLY-BLANK-VOUCHER-REGISTER-TAPES;
        ASSIGN TOTAL,ENTRY-DATE OF VOUCHER-REGISTER-TAPE;
        TRANSFER VOUCHER-REGISTER-TAPE TO GENERAL-LEDGER;
        PUT VOUCHER-REGISTER-TOTAL INTO VOUCHER-REGISTER;
        END TASK;
    END CLERK3.

    DEPT-HEAD:
        WAIT FOR VOUCHER1,VOUCHER2,REQ3,PURCHASE-ORDER4,RR3,INVOICE1;
        ASSIGN APPROVAL OF VOUCHER1;
        ASSIGN APPROVAL OF VOUCHER2;
        TRANSFER VOUCHER1,VOUCHER2,REQ3,PURCHASE-ORDER4,RR3,INVOICE1
        TO ACCOUNT-PAYABLE.CLERK2;
        END TASK;
    END DEPT-HEAD.
    END ACCOUNT-PAYABLE.

    CASH-DISBURSEMENTS:
    CLERK1:
        WAIT FOR VOUCHER1,REQ3,PURCHASE-ORDER4,RR3,INVOICE1;
        IF PURCHASE-ORDER4.VENDOR <> RR3.VENDOR
        OR PURCHASE-ORDER4.VENDOR <> INVOICE1.VENDOR
        OR PURCHASE-ORDER4.VENDOR <> VOUCHER1.VENDOR THEN
            REVIEW;
        END TASK;
        END IF;
        IF PURCHASE-ORDER4.PRODUCT-DESC <> REQ3.PRODUCT-DESC
        OR PURCHASE-ORDER4.PRODUCT-DESC <> RR3.PRODUCT-DESC
        OR PURCHASE-ORDER4.PRODUCT-DESC <> INVOICE1.PRODUCT-DESC THEN
            REVIEW;
        END TASK;

```

Figure 3 (Cont.)

```

END IF;
IF PURCHASE-ORDER4.PRODUCT-NUMBERS <> REQ3.PRODUCT-NUMBERS
OR PURCHASE-ORDER4.PRODUCT-NUMBERS <> RR3.PRODUCT-NUMBERS
OR PURCHASE-ORDER4.PRODUCT-NUMBERS <> INVOICE1.PRODUCT-NUMBERS THEN
REVIEW;
END TASK;
END IF;
IF PURCHASE-ORDER4.PRICE <> INVOICE1.PRICE THEN
REVIEW;
END TASK;
END IF;
IF PURCHASE-ORDER4.QTY <> REQ3.QTY
OR PURCHASE-ORDER4.QTY <> RR3.QTY
OR PURCHASE-ORDER4.QTY <> INVOICE1.QTY THEN
REVIEW;
END TASK;
END IF;
IF VOUCHER1.AMOUNT <> INVOICE1.QTY * INVOICE1.PRICE THEN
REVIEW;
END TASK;
END IF;
IF VOUCHER1.AMOUNT > 9999.99 AND VOUCHER1.APPROVAL <> OK THEN
REVIEW;
END TASK;
END IF;
ASSIGN CANCELLATION-FIELD OF PURCHASE-ORDER4;
ASSIGN CANCELLATION-FIELD OF REQ3;
ASSIGN CANCELLATION-FIELD OF INVOICE1;
ASSIGN CANCELLATION-FIELD OF VOUCHER1;
TRANSFER VOUCHER1 TO CASH-DISBURSEMENTS.CLERK2;
PUT REQ3,PURCHASE-ORDER4,RR3,INVOICE1 INTO
PURCHASE-HISTORY-FILE;
END TASK;
END CLERK1.

CLERK2:
WAIT FOR VOUCHER1;
GET CHECK FROM BLANK-CHECK-SUPPLY;
ASSIGN PAYEE,DATE-PAID,AMOUNT,AUTHORIZATION OF CHECK;
GET CHECK-REGISTER-ENTRY FROM SUPPLY-BLANK-CHECK-REGISTER-ENTRY;
ASSIGN CHECK-NUMBER,PAYEE,ENTRY-DATE,AMOUNT OF CHECK-REGISTER-ENTRY;
PUT CHECK-REGISTER-ENTRY INTO CHECK-REGISTER;
GET CHECK-REGISTER-TOTAL FROM CHECK-REGISTER;
ASSIGN TOTAL,ENTRY-DATE OF CHECK-REGISTER-TOTAL;
PUT CHECK-REGISTER-TOTAL INTO CHECK-REGISTER;
GET ACCOUNT-PAID-NOTIFICATION FROM SUPPLY-BLANK-NOTIFICATIONS;
ASSIGN VOUCHER-NUMBER,CHECK-NUMBER OF ACCOUNT-PAID-NOTIFICATION;
TRANSFER ACCOUNT-PAID-NOTIFICATION TO ACCOUNT-PAYABLE.CLERK3;
PUT VOUCHER1 INTO PURCHASE-HISTORY-FILE;
TRANSFER CHECK TO VENDOR;
END TASK;

MONTHLY GET CHECK-REGISTER-TOTAL FROM CHECK-REGISTER;
GET CHECK-REGISTER-TAPE FROM SUPPLY-BLANK-CHECK-REGISTER-TAPES;
ASSIGN TOTAL,ENTRY-DATE OF VOUCHER-REGISTER-TAPE;
TRANSFER CHECK-REGISTER-TAPE TO GENERAL-LEDGER;
PUT CHECK-REGISTER-TOTAL INTO CHECK-REGISTER;
END TASK;
END CLERK2.
END CASH-DISBURSEMENTS.

```

Figure 3 (Cont.)

VENDOR:  
 WAIT FOR CHECK;  
 TRANSFER CHECK TO BANK;  
 END TASK;  
 END VENDOR.

BANK:  
 WAIT FOR DEPOSIT-SLIP1,DEPOSIT-SLIP2;  
 PUT DEPOSIT-SLIP1,DEPOSIT-SLIP2 INTO DEPOSIT-SLIP-FILE;  
 END TASK;  
 WAIT FOR CHECK;  
 ASSIGN CANCELLATION-FIELD,DATE-CASHED OF CHECK;  
 PUT CHECK INTO CANCELLED-CHECK-FILE;  
 END TASK;  
 MONTHLY GET DEPOSIT-SLIP1(S) FROM DEPOSIT-SLIP-FILE;  
 GET CHECK(S) FROM CANCELLED-CHECK-FILE;  
 GET BANK-STATEMENT FROM BANK-STATEMENT-SUPPLY;  
 ASSIGN CHECK-NUMBER,DATE-CASHED,CHECK-AMT,DEPOSIT-NUMBER,  
 DATE-OF-DEPOSIT;DEPOSIT-AMT OF BANK-STATEMENT;  
 TRANSFER BANK-STATEMENT,DEPOSIT-SLIP1(S),CHECK(S) TO AUDITOR;  
 END TASK;  
 END BANK.

PURCHASE-EXPENSE-LEDGER:  
 CLERK1:  
 WAIT FOR VOUCHER2;  
 GET PURCHASE-EXPENSE-JOURNAL-ENTRY FROM  
 SUPPLY-BLANK-PURCHASE-EXPENSE-JOURNAL-ENTRIES;  
 ASSIGN DATE,ACCOUNT-NUMBER,ACCOUNT-DESC,AMOUNT,TYPE OF  
 PURCHASE-EXPENSE-JOURNAL-ENTRY;  
 PUT PURCHASE-EXPENSE-JOURNAL-ENTRY INTO PURCHASE-EXPENSE-JOURNAL  
 GET PURCHASE-EXPENSE-LEDGER-ENTRY FROM  
 SUPPLY-BLANK-PURCHASE-EXPENSE-LEDGER-ENTRIES;  
 ASSIGN ACCOUNT-NUMBER,ACCOUNT-DESC,ENTRY-DATE,AMOUNT,TYPE OF  
 PURCHASE-EXPENSE-LEDGER-ENTRY;  
 PUT PURCHASE-EXPENSE-LEDGER-ENTRY INTO PURCHASE-EXPENSE-LEDGER;  
 GET PURCHASE-EXPENSE-LEDGER-TOTAL FROM PURCHASE-EXPENSE-LEDGER;  
 ASSIGN ACCOUNT-NUMBER,NET-TOTAL,GRAND-NET-TOTAL,ENTRY-DATE  
 OF PURCHASE-EXPENSE-LEDGER-TOTAL;  
 PUT PURCHASE-EXPENSE-LEDGER-TOTAL INTO PURCHASE-EXPENSE-LEDGER;  
 PUT VOUCHER2 INTO PURCHASE-EXPENSE-VOUCHER-FILE;  
 END TASK;  
 MONTHLY GET PURCHASE-EXPENSE-LEDGER-TOTAL FROM PURCHASE-EXPENSE-LEDGER;  
 GET PURCHASE-EXPENSE-LEDGER-TAPE FROM  
 SUPPLY-BLANK-PURCHASE-EXPENSE-LEDGER-TAPE;  
 ASSIGN ENTRY-DATE,ACCOUNT-NUMBER,NET-TOTAL,GRAND-NET-TOTAL OF  
 PURCHASE-EXPENSE-LEDGER-TAPE;  
 TRANSFER PURCHASE-EXPENSE-LEDGER-TAPE TO GENERAL-LEDGER;  
 PUT PURCHASE-EXPENSE-LEDGER-TOTAL INTO PURCHASE-EXPENSE-LEDGER;  
 END TASK;  
 END CLERK1.  
 END PURCHASE-EXPENSE-LEDGER.

CASH-RECEIPTS:  
 GET DEPOSIT-SLIP1 FROM SUPPLY-BLANK-DEPOSIT-SLIPS;  
 ASSIGN ACCOUNT-NUMBER,DATE-OF-DEPOSIT,DEPOSITOR,AMOUNT OF DEPOSIT-SLIP1;

Figure 3 (Cont.)

```
COPY DEPOSIT-SLIP1 GIVING DEPOSIT-SLIP2;
TRANSFER DEPOSIT-SLIP1,DEPOSIT-SLIP2 TO BANK;
END TASK;
END CASH-RECEIPTS.

GENERAL-LEDGER:
WAIT FOR VOUCHER-REGISTER-TAPE,CHECK-REGISTER-TAPE,
PURCHASE-EXPENSE-LEDGER-TAPE;
IF CHECK-REGISTER-TAPE.TOTAL <> VOUCHER-REGISTER-TAPE.TOTAL THEN
REVIEW;
END TASK;
END IF;
GET GENERAL-LEDGER-ENTRY FROM SUPPLY-BLANK-GENERAL-LEDGER-FORMS;
ASSIGN ACCOUNT-NUMBER,ACCOUNT-DESC,ENTRY-DATE,AMOUNT,TYPE
OF GENERAL-LEDGER-ENTRY;
PUT GENERAL-LEDGER-ENTRY INTO GENERAL-LEDGER;
PUT VOUCHER-REGISTER-TAPE, CHECK-REGISTER-TAPE,
PURCHASE-EXPENSE-LEDGER-TAPE INTO TAPE-HISTORY-FILE;
END TASK;
END GENERAL-LEDGER.

AUDITOR:
WAIT FOR BANK-STATEMENT,DEPOSIT-SLIP1(S),CHECK(S);
PUT BANK-STATEMENT,DEPOSIT-SLIP1(S),CHECK(S) INTO AUDIT-FILE;
END AUDITOR.
END SYSTEM OPERATIONS.

TASK ASSIGNMENTS:
ACCOUNT-PAYABLE.CLERK2 IS 839,963;
ACCOUNT-PAYABLE.CLERK3 IS 109,738,105;
CASH-DISBURSEMENTS.CLERK1 IS 444,453;
CASH-DISBURSEMENTS.CLERK2 IS 678,934;
PURCHASE-EXPENSE-LEDGER.CLERK1 IS 554;
END TASK; ASSIGNMENTS.
END SYSTEM.
```

Figure 3 (Cont.)



Figure 4  
General-Ledger

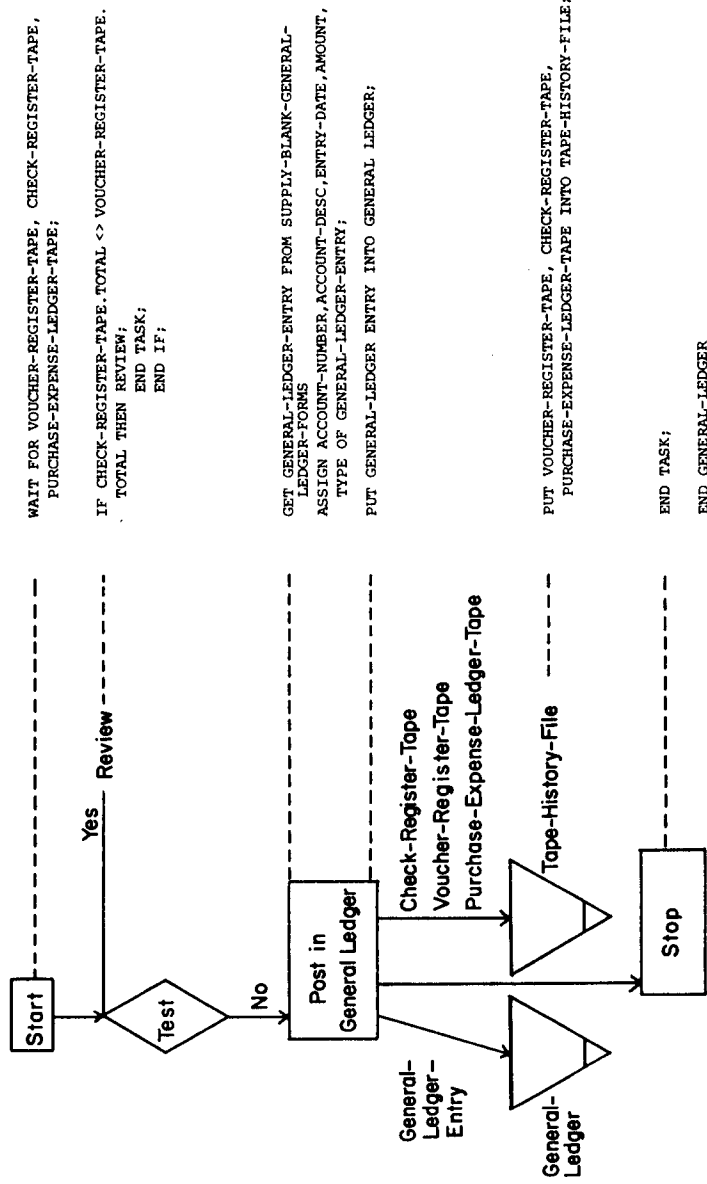


Figure 5

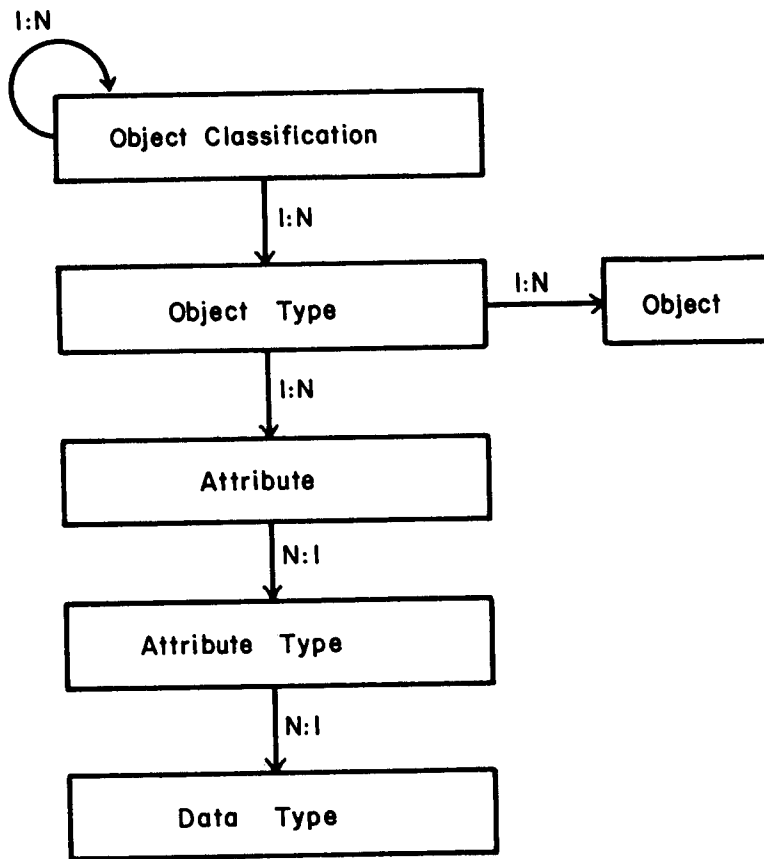
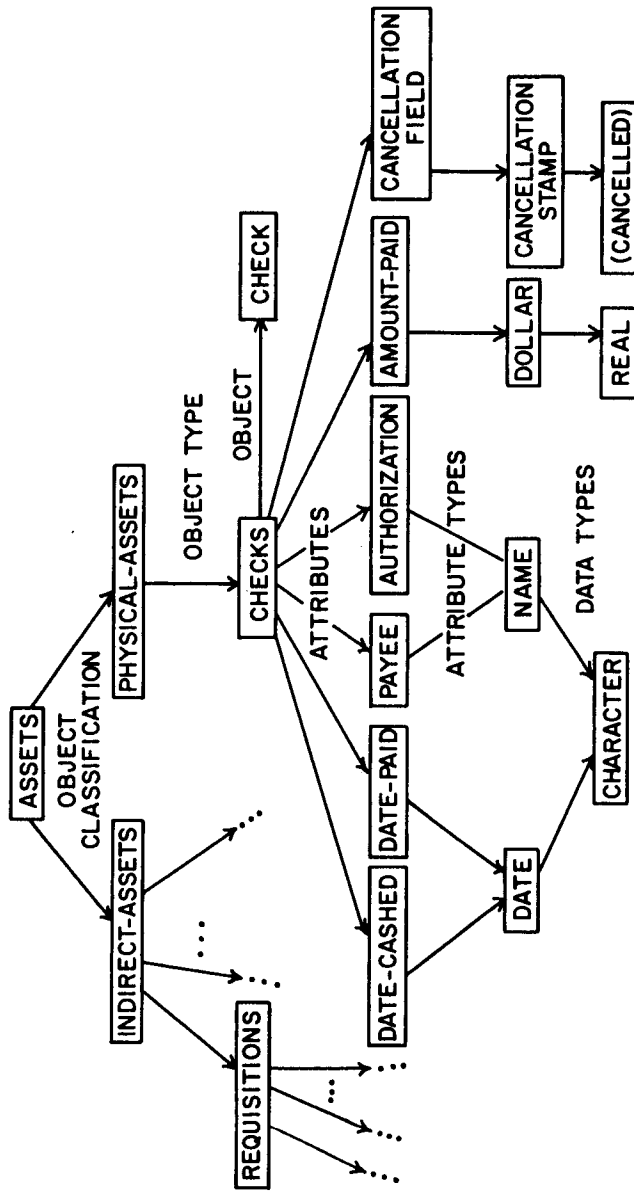


Figure 6



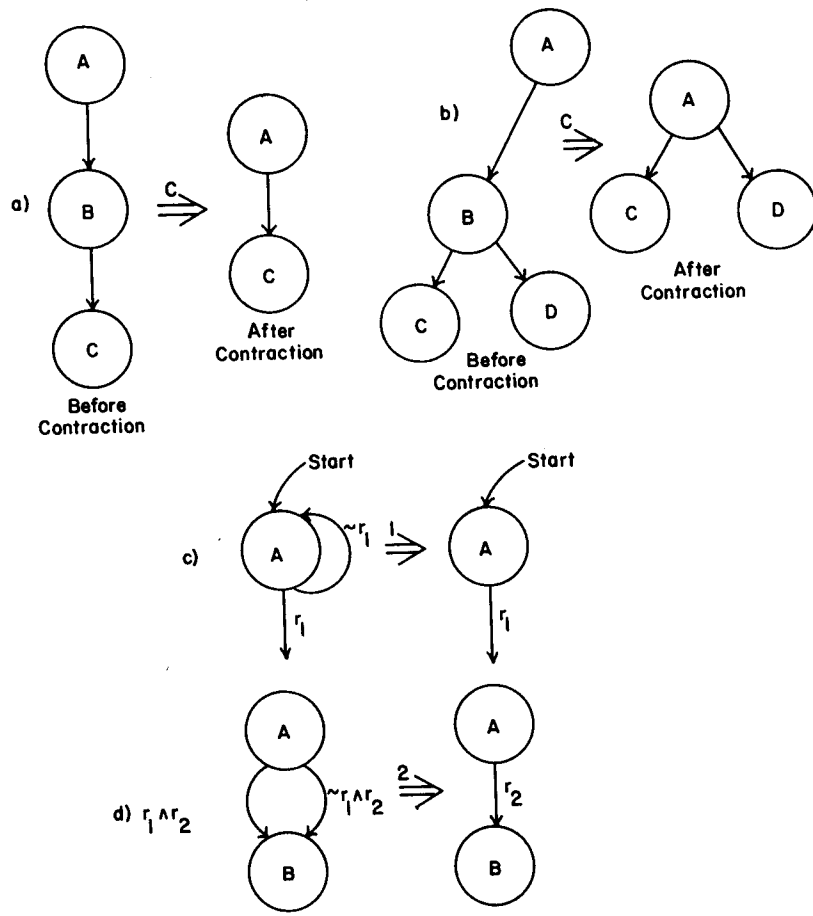


Figure 7

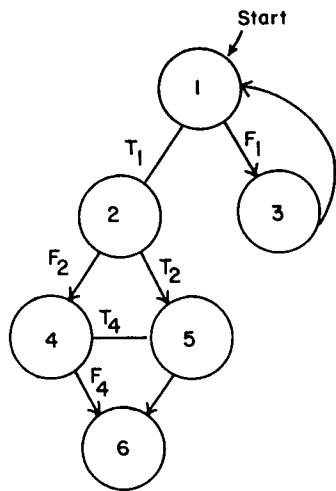


Figure 8  
Original System

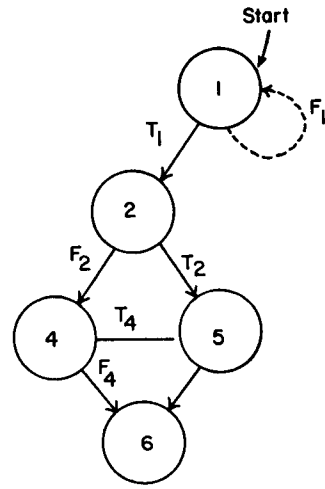


Figure 9  
Contract out node 3 and removal  
of arc  $F_1$  using rule 1

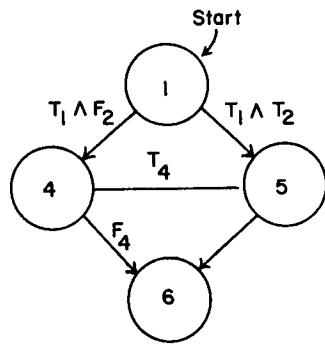


Figure 10  
Contract out node 2

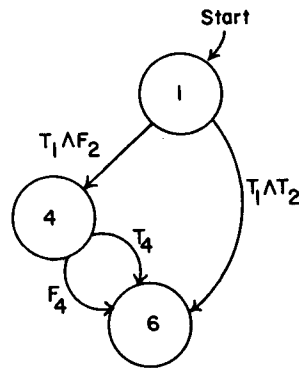


Figure 11  
Contract out node 5

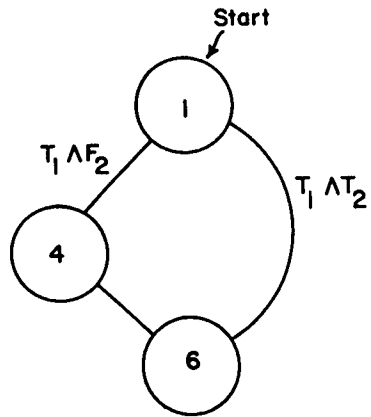


Figure 12  
Arc 4 6 cleanup using rule 2

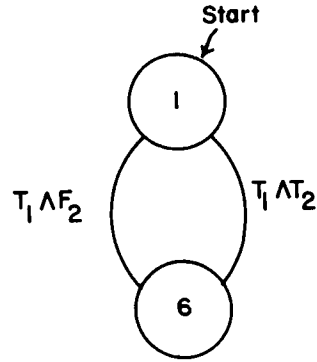


Figure 13  
Contract out node 4

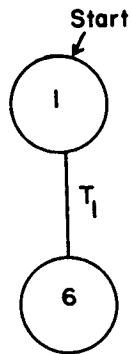


Figure 14  
Arc 1 6 cleanup using rule 2

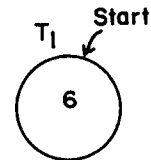


Figure 15  
Contract out node 1

## References

- AICPA. 1972. Statements on Auditing Standards. American Institute of Certified Public Accountants, AU Section 320, November.
- \_\_\_\_\_. 1980. Report of the Special Advisory Committee on Internal Control. New York: AICPA.
- Arthur Andersen & Co. 1978. A Guide for Studying and Evaluating Internal Accounting Controls.
- Bailey, A. D., Jr.; Gerlach, J; McAfee, R. P.; and Whinston, A. B. 1980a. "An Application of Complexity Theory to the Analysis of Internal Control Systems." Working paper, Purdue University, July.
- \_\_\_\_\_. 1980b. "TICOM II: Theory." Working paper, Purdue University, July.
- \_\_\_\_\_. In press. "Office Automation." In Handbook of Industrial Engineering, Salvendy, ed. New York: John Wiley & Sons, Inc.
- Deloitte Haskins & Sells. 1979. Internal Accounting Control, An Overview of the DH & S Study and Evaluation Techniques.
- Ellis, C., and Morris, P. 1979. "Office Streamlining." IRIA. Rocquencourt, France, Nov. 6-9.
- Griffith, Samuel B. 1963. Sun Tzu: The Art of War. New York: Oxford University Press.
- Jensen J., and Wirth, N. 1975. PASCAL User Manual and Report. New York: Springer-Verlag.
- Robertson, J. C. 1979. Instructor's Guide to Accompany Auditing, rev. ed. Houston, Tex.: Business Publications, Inc.

A Comment on Bailey et al.'s  
"TICOM II: The Internal Control Language"

Miklos A. Vasarhelyi  
Columbia University

For a long time I have been a great admirer of the Bailey et al. efforts in the automation of internal control design and evaluation procedures. I would like to start my discussion by classifying their paper within a taxonomy of research efforts that I have been developing in the recent months (see Vasarhelyi and Berk, 1981). Their paper uses, as a research method, the "analytical-internal logic" approach; no data are utilized. Its inference style is "inductive," because no hypotheses are developed or tested. Its mode of reasoning is "mixed," using both analytical and qualitative arguments. Its mode of analysis is "normative," advocating the use of a particular approach. In addition, it could be broadly classified (1) in the "Statistical Modeling-Other" school of thought, as it will rely on internal control modeling, (2) in the "audit" accounting area, (3) as applicable in the "long term," (4) as using "mathematics and engineering" as the foundation discipline, and (5) as information and treatment that could be vaguely classified as "internal information--internal controls" and "measurement." My difficulties with the classification of this paper in accounting literature are symptomatic of the difficulties I have always had with TICOM papers. They are very difficult to read, and their message is not very clear. I will omit any discussion of the "computerized office information systems," often mentioned but never explored in the paper; the linkage with the topic is obvious, but it is not well illustrated and it is not warranted considering the state of the art of the TICOM system. The reader interested in the office automation implications should refer to Bailey et al. (in press).

Reading this paper is very difficult. In my opinion the authors could have attempted two approaches. One would show a very simple example and explain it thoroughly to the reader, making clear the key features of the system. This approach would not prove the applicability of the system but would illustrate to the reader what is being



attempted. The second is the one attempted by Bailey et al., in which they try to illustrate the applicability of TICOM II to a realistic situation. Unfortunately, there they did not take it to a successful completion. Despite nearly 500 lines of Pl/1-like code, a successful execution was not shown. The system was not yet successfully implemented in terms of software, and it seems that this would be an essential step for this paper.

The paper deals with many important issues but at only a superficial level. Among these are the heuristic resolution of nets, analytical formulation of internal controls, inconsistency checking of internal controls, perpetual audit issues, and office automation questions. The paper should have been restricted to one or two of these issues thoroughly explored.

However, if internal control systems can be succinctly and unambiguously described, a contribution is made to the state of the art. It must also be shown that this type of description is preferable to the traditional one.

The main objective of the paper is the computer analysis of the internal control system. This should not be an objective per se. An objective should be the formalization of internal control systems, the automatic detection of internal control inconsistencies, or improvements in the economics of design or implementation of computer systems. The computer and/or analytical processes would be a tool in this process.

TICOM II seems to have four main elements. The first allows the representation of an internal control system as a formalized (computerized) set of descriptors. The second serves to synthesize the system, integrate, and scan for logical errors. The third is a query system by which the auditor may interactively search for internal control weaknesses or test control features. The fourth is denominated "a monitor which permits the interfacing of the control evaluation system with the actual events of the office." This last element seems to indicate a feature by which TICOM would be used to actually design and implement controls in the organization. The paper does not explain what elements of the system are currently functional.

TICOM II seems to be the general name for the methodology and its four parts described above. More specifically it contains a computer

language denominated ICDL (Internal Control Description Language) which has as its most basic elements objects, processors, repositories, and transactions.

Objects are the elements to be processed by the system (e.g., checks, goods, and data records). Processors represent people or work-related positions. Repositories are the containers for objects, and transactions move objects from repository to repository.

In order to write an ICDL system description, three main steps must be followed: objects and repositories must be described, system operations specified, and tasks assigned.

Figure 2 of their paper lists some of the ICDL commands that control the flow of transactions. Figure 3 illustrates an ICDL program. The initial lines define objects and repositories. Once definitions are completed the system operations are described. The final element of figure 3 is the task assignments. The authors are mute about actual runs and interaction with the TICOM II system and the described model.

Leaving aside specific details, I would like to conclude with six points.

- (1) The paper was somewhat premature; the software should have been substantially completed before this paper was written.
- (2) The auditor needs more help in the test of compliance where the key problems related to internal control evaluation rest. TICOM II has not yet been used to address these problems.
- (3) Manual procedures of internal control description (flowcharts) seem to be adequate and easy to learn. The paper fails to show the advantage of TICOM II in these two areas. TICOM, if fully implemented, would probably be somewhat interchangeable with flowcharts. The cost-benefit trade-offs between the two methods cannot be estimated at this point. TICOM will probably be able to better detect certain types of logical inconsistencies than traditional flowcharts but would do little in relation to separation of duties.
- (4) Another key issue of the auditor relates to the efficiency and efficacy of internal controls. These are not discussed in the paper, and they may entail the main advantage of a system like the one proposed. In such a system the software and formal system description

would be used as a model of the real system and different conditions simulated.

Also on this point, a system for evaluating internal controls must deal with the problems of serial, parallel, and overlapping controls. DH&S developed its concept of "critical control combinations" to deal with this problem.

(5) TICOM may, on the other hand, work within the context of current audit procedures by providing an environment where (a) internal control documentation is standardized (if allowable variable names are restricted to a set), (b) ICQ's are automatically tied to these standardized descriptors, (c) standardized internal control programs exist by industry and are used as evaluating benchmarks, and (d) the system is tied in to traditional audit objectives and procedures.

(6) This approach is deserving of research and exploration, but it still needs much work before it can be evaluated as a useful audit tool.

## References

- Bailey, A. D.; Gerlach, J.; McAfee, R. P.; and Whinston, A. B. In press. "Office Automation." In Handbook of Industrial Engineering, ed. Salvendy. New York: John Wiley & Sons, Inc.
- Vasarhelyi, M. A., and Berk, J. 1981. "The Multiple Taxonomies of Accounting Research." Working paper, Columbia University.